What is claimed is:

1. A method for encrypting at least one byte of plaintext to produce at least one byte of ciphertext, the method comprising:

selecting a secret key to create an S-vector following a standard encryption method;

setting a sequence number, the sequence number having a first part and a second part;

setting a first variable as the first part of the sequence number;

setting a second variable as the second part of the sequence number;

setting a byte sequence number;

calculating a third variable as the sum of the second variable plus the byte sequence number;

incrementing the byte sequence number by one;

calculating a fourth variable by adding the first variable plus the value within the S-vector pointed to by the third variable;

locating an encryption byte, wherein the location of the encryption byte within the S-vector is pointed to by the sum of the value within the S-vector pointed to by the third variable plus the value within the S-vector pointed to by the fourth variable; and

exclusive ORing the encryption byte with the at least one byte of plaintext to generate the at least one byte of ciphertext.

2. The method of claim 1 where setting a second variable further comprises:

exclusive ORing the second part of the sequence number and the value within the S-vector pointed to by the first variable.

3. The method of claim 1, wherein calculating a fourth variable further comprises:

setting a value for a counter; and

summing the first variable plus the value within the S-vector pointed to by the third variable plus the value within the S-vector pointed to by the counter.

4. A method of encrypting one or more packets of plaintext, the one or more packets of plaintext having a plurality of bytes of plaintext, the method comprising:

obtaining a secret key;

generating an S-vector using the secret key;

for each successive one or more packets of plaintext,

obtaining a sequence number having a first portion and a second portion;

setting a first variable using the first portion of the sequence number;

setting a second variable using the second portion of the sequence number; and

setting a byte sequence number equal to zero;

for each next byte of the plurality of bytes of plaintext, calculating a next encryption byte, the calculating comprising:

adding the second variable to the byte sequence number to produce a third variable;

calculating a fourth variable by adding the first variable plus the value within the S-vector pointed to by the third variable;

locating a next encryption byte within the S-vector by adding the values within the S-vector pointed to by the third variable and the fourth variable to calculate a pointer to locate the next encryption byte;

setting the second variable equal to the third variable; and

incrementing the byte sequence number by one.

5. The method of claim 4 wherein calculating a second variable comprises:

exclusive ORing the second portion of the sequence number with the value within the S-vector pointed to by the first variable.

6. The method of claim 4, at the transmitter further comprising:

for each next encrypted byte, calculating a next ciphertext byte by XORing the next encryption byte with the next byte of the plurality of bytes of plaintext.

7. The method of claim 4, at the receiver further comprising:

for each next encryption byte, calculating a received next plaintext byte by XORing the next encryption byte with the next ciphertext byte within each one or more packets of plaintext.

8. The method of claim 4, wherein calculating a fourth variable further comprises:

setting a counter;

calculating the fourth variable by adding the first variable plus the values within the S-vector pointed to by the third variable and the counter; and

for each next one or more packets of plaintext, incrementing the value of the counter according to a predetermined schedule.

9. The method of claim 8, wherein setting a counter further comprises:

for a first packet of the one or more packets of plaintext, resetting a rollover counter to zero;

for each next one or more packets of plaintext, incrementing the rollover counter when incrementing the sequence number causes the value of the sequence number to transition from to all 0s.

10. The method of claim 4 further comprising:

for each next byte of the plurality of bytes of plaintext, permuting the S-vector, the permutation comprising:

saving a copy of the S-vector; and

swapping the value within the S-vector pointed to by the third variable and the value within the S-vector pointed to by the fourth variable, wherein the values within the S-vector are swapped after locating the next encryption byte; and

for each next one or more packets of plaintext, restoring the saved S-vector.

11. A method for converting one or more packets having a plurality of bytes of plaintext P to one or more packets having a plurality of ciphertext bytes C, the method comprising:

obtaining a secret key;

calculating an S-vector having a plurality of S-vector bytes using the secret key;

randomly setting a sequence number having a first part and a second part;

for each successive one or more packets, incrementing the sequence number;

setting a first variable j according to j = first part of the sequence number;

calculating a second variable i according to i = second part of the sequence number;

for each successive byte of the plurality of bytes of plaintext P, calculating a next successive ciphertext byte C, the calculating comprising:

further calculating the first variable according to j = j + S[i];

setting a third variable k;

further calculating the second variable i according to i = i + k;

locating the next successive encryption byte E within the S-vector according to E = S[S[i] + S[j]]; and

converting the next successive encryption byte E to a next successive ciphertext byte C according to C = E ⊕ P; and

when the last byte of the plurality of bytes of plaintext P has been converted to ciphertext byte C for the next packet of the one or more packets, transmitting the next successive packet of the one or more packets to a receiver.

12. The method of claim 11 wherein calculating a second variable i further comprises:

exclusive ORing the low order sequence number and the value within the S-vector pointed to by first variable according to i = (low order of the sequence number) ⊕ S[j].

13. The method of claim 11, wherein further calculating the first variable j further comprises:

setting a counter r;

further calculating the first variable j according to j = j + S[i] + S[r]; and

for each successive packet of the one or more packets, incrementing the value of the counter r.

14. The method of claim 11, wherein calculating a next successive encryption byte E further comprising:

permuting the S-vector, the permutation comprising:

saving a copy of the S-vector; and

swapping the byte of the plurality of S-vector bytes pointed to by the first variable j and the byte of the plurality of S-vector bytes pointed to by the second variable i; and

when the last byte of the plurality of bytes of plaintext P has been converted to the plurality of ciphertext bytes C for the next one of the one or more packets, restoring the saved S-vector.

15. A method for converting one or more packets having a plurality of bytes of plaintext P to one or more packets having a plurality of ciphertext bytes C, the method comprising:

obtaining a secret key;

calculating an S-vector having a plurality of S-vector bytes using the secret key;

randomly setting a sequence number having a high order and a low order;

for each successive one or more packets, incrementing the sequence number;

for each successive byte of the plurality of bytes of plaintext P, calculating a next successive encryption byte E, the calculating comprising:

setting a first variable j according to j = high order of the sequence number;

calculating a second variable i according to i = (low order of the sequence number) $\oplus$ S[j];

setting a counter r;

further calculating the first variable according to $j = j + S[i] + S[r]$;

setting a third variable k;

incrementing the second variable i according to $i = i + k$;

locating the next successive encryption byte E within the S-vector according to $E = S[\ S[i] + S[j]\ ]$; and

converting the next successive encryption byte E to a next successive ciphertext byte C according to $C = E \oplus P$; and

when the last byte of the plurality of bytes of plaintext P has been converted to ciphertext byte C for the next packet of the one or more packets, transmitting the next successive packet of the one or more packets to a receiver.